

ALGORITMOS PARA LA PLANIFICACIÓN Y SEGUIMIENTO DE TRAYECTORIAS EN ROBOTS AGRICOLAS

CHRISTIAN CORREA ¹, LORENZO VÁSQUEZ ²

¹ Ingeniero Civil Agrícola, M.Sc., Dr ©, Profesor Asistente, Facultad de Ing. Agrícola, Dpto. de Mecanización y Energía, Universidad de Concepción, Chile, ccorrea@udec.cl.

² Ingeniero Civil Eléctrico, M.Sc. ©. Profesor Instructor. Universidad de Tarapacá, Chile. lvasquez@uta.cl

Apresentado no
X Congreso Latinoamericano y del Caribe de Ingeniería Agrícola - CLIA 2012
XL Congresso Brasileiro de Engenharia Agrícola - CONBEA 2012
15 a 19 de julho de 2012 - Londrina - PR, Brasil

RESUMEN: La base para realizar cualquier tarea agrícola mediante robots, es la planificación y seguimiento de rutas o trayectorias. Así, el objetivo de esta investigación es desarrollar e implementar algoritmos de seguimiento y planificación (global y local) de trayectorias de robots agrícolas. La planificación global se realizó mediante el algoritmo A* aplicado sobre mapas de cultivo y la planificación local se realizó aplicando A* sobre un mapa 2D obtenido a partir de imágenes 3D de los obstáculos encontrados en el camino. En cuanto el seguimiento de trayectorias, esta se realizó implementando una aproximación numérica de la trayectoria mediante el método de Euler. Los parámetros correspondientes a la dinámica del controlador de la trayectoria del robot fueron obtenidos mediante algoritmos genéticos. El mapa 3D fue generado a partir del sensor Kinect de Microsoft y sus datos procesados usando Matlab 2010b. Los resultados preliminares muestran que es posible implementar estos algoritmos en pequeños robots diseñados para cultivos hilerados. Proveyendo así, una metodología robusta que permite seguir las rutas asignadas con errores inferiores a RMSE=0.1m en trayectorias de 30m.

PALABRAS CLAVE: Planificación, trayectorias, robot móvil.

ALGORITHMS, FOR PATH FOLLOWING AND PLANNING, FOR AGRICULTURAL ROBOTS.

ABSTRACT: The basis to perform any agricultural task by robots is to planning and following paths or trajectories inside the crops. This research aims to developed and implemented algorithms for following and planning (global and local) trajectories for agricultural robots. The global planning was performed using the A * algorithm applied over crop maps and the local planning was performed using A* applied over a 2D map obtained from 3D images of the obstacles present on the way. Trajectory following, was done by implementing a numerical approximation of the trajectory by Euler's method. The parameters for the dynamics of the robot trajectory's controller were obtained by genetic algorithms. The 3D map was generated from the Microsoft Kinect sensor, and its data processed by Matlab 2010b. Preliminary results show that these algorithms can be implemented in small robots designed to be used in crop rows. Thereby providing a robust methodology to following assigned paths with errors less than RMSE= 0.1 m in trajectories of 30 m.

KEYWORDS: Path Planning, tracking, mobile robot.

INTRODUCCION

Un robot móvil necesita tener un completo entendimiento del terreno y características circundantes para estar en condiciones de navegar de forma segura en entornos complejos (SOFTMAN *et al.*, 2006). Razón por la cual se hace imprescindible dotarlo de una trayectoria segura a seguir, así como de sensores que le permitan detectar y evadir los obstáculos que se le presenten.

En esta investigación, se propone una serie de algoritmos que apuntan a resolver el problema de eliminación de malezas en cultivos hilerados. En particular abordaremos la planificación de las trayectorias a seguir, la generación de curvas que aproximen dichas trayectorias, la implementación y optimización de un controlador de la trayectoria y finalmente la evasión de obstáculos presentes en el camino.

MATERIALES Y MÉTODOS

La metodología la dividiremos en tres secciones: 1) Planificación de trayectoria con A*. 2) Suavizado de la trayectoria mediante curvas de Bézier. 3) Modelo cinemático del robot. 4) Algoritmo de seguimiento de trayectoria.

1) Planificación de trayectoria con A*.

El algoritmo A* propuesto por HART et al. (1968), es un algoritmo que busca la ruta más corta (de menor costo) entre dos puntos. Es uno de los algoritmos de planificación más usados (FLETCHER *et al.* 2008), debido a que de haber una solución, la encuentra y garantiza que es óptima.

En el escenario propuesto, el robot debe ir desde su posición actual, hasta el punto con maleza más cercano. La planificación de este recorrido la denominamos Planificación Global.

- a) **Planificación Global.** Supóngase que el cultivo es dividido en una cuadrícula, y que el robot es de tamaño menor o igual que una casilla de esta cuadrícula. El Algoritmo A* se encarga de encontrar la ruta más corta, entre la posición actual y el punto de maleza más cercano (Figura 1a).

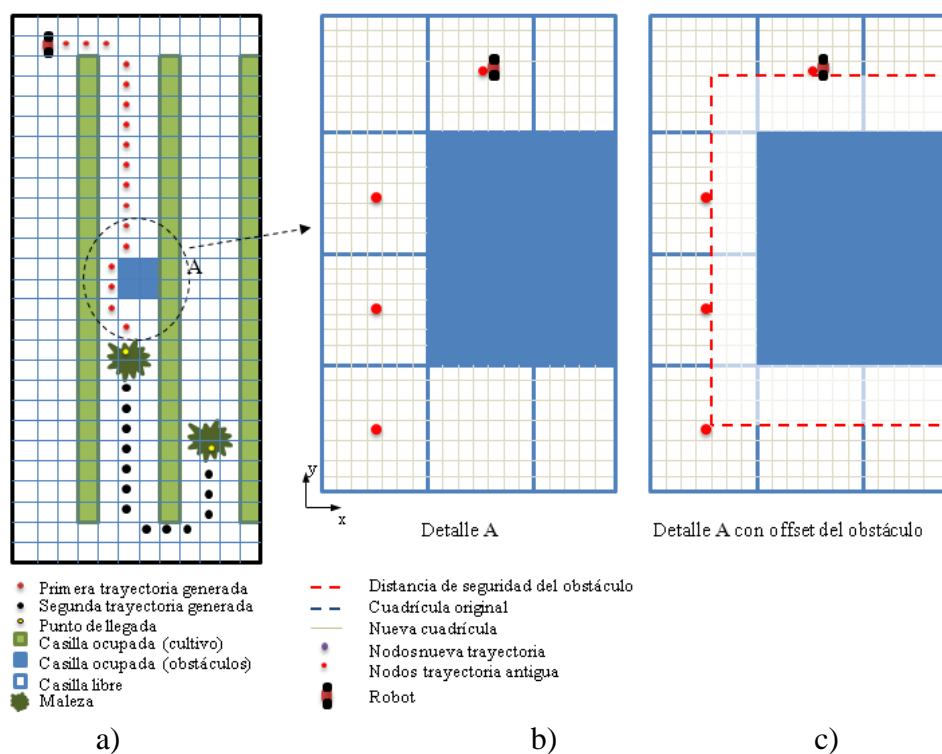


FIGURA 1. a) Trayectorias generadas por el algoritmo A*. b) Detalle de la trayectoria global usada para evitar un obstáculo conocido. c) Generación del offset.

En la Figura 1a, se puede observar en puntos de color rojo, la trayectoria que sigue el robot desde su ubicación inicial al punto más cercano que contiene maleza. Luego toma ese punto como punto de partida y genera una nueva trayectoria, marcada con puntos de color negro, hasta el punto más cercano que contiene maleza. Este es un proceso cíclico en el que avanza desde un punto con maleza hasta el siguiente.

- b) **Planificación Local.** Para evitar los obstáculos inesperados presentes en la ruta, el robot extrae información del mapa del entorno 3D, generada por el sensor Kinect de Microsoft (Figura 2a). El mapa 3D es adquirido empleando el Microsoft Kinect Windows SDK. A modo de interface con el modulo de planificación, se desarrolló una rutina en Matlab que se comunica con el SDK.

El proceso de extracción de información desde el mapa 3D, consiste en considerar que todo objeto de más de 8 cm de altura, es un obstáculo a evitar. De este modo, cada casilla de la grilla que contenga in objeto con valores de $z > 8$ cm, se considera una casilla ocupada (Figura 2d).

A estos obstáculos se les asigna una dimensión, en el plano xy , mayor a la real (la mitad del ancho del robot), con el propósito de evitar colisiones. En otras palabras, se hace un offset del obstáculo (Figura 2e).

Una vez que se tiene el mapa del entorno y los obstáculos en el, se genera una nueva trayectoria desde la posición actual a la de destino. Esta vez con una grilla de mayor densidad, con el propósito de generar suficientes puntos intermedios, que garanticen evitar la colisión con el obstáculo (Figura 2e).

Cuando el robot se encuentra con objetos que no están en su mapa, cambia al modo evasión de obstáculos. Este modo, genera un mapa de dos dimensiones en el plano xy .

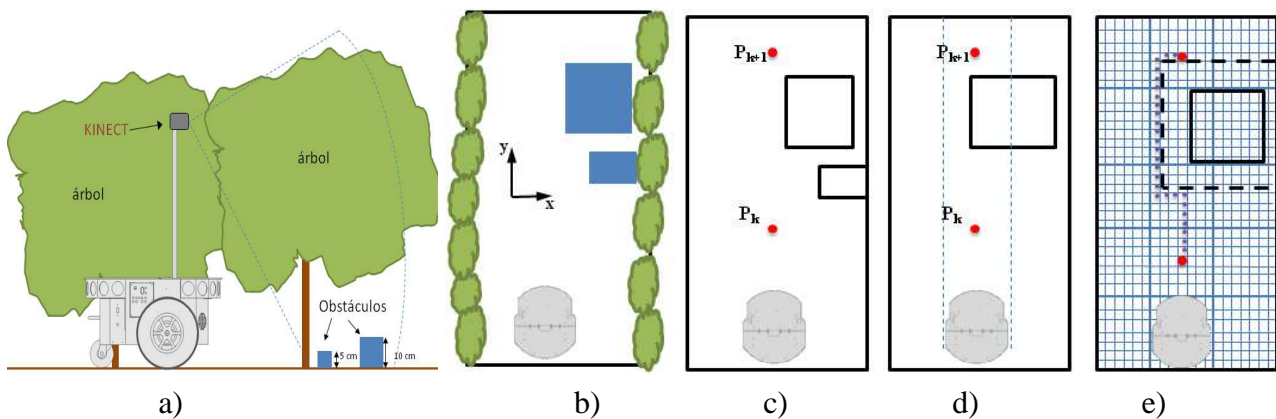


FIGURA 2. a) Robot y sensor Kinect. b) Vista plano xy . c) Síntesis del plano xy . d) Elimina objetos menores a 8 cm y decide si puede o no seguir. e) Offset de objetos y nueva trayectoria.

2) Suavizado de la trayectoria mediante curvas de Bézier.

Una vez que se dispone de los puntos (generados por A*) por los que el robot debe pasar, estos son unidos mediante curvas de Bézier implementadas mediante el algoritmo de Casteljau como se describe en CHOI et al. (2008).

Las curvas de Bézier son paramétricas y diseñadas para unir suavemente un grupo de puntos. Estas curvas pueden ser de diversos grados, dependiendo del nivel de suavizado que se desee.

El requerimiento mínimo en planificación de trayectorias, es pasar por el punto de inicio y por el final con una dirección específica. Así, el mínimo grado de la curvas de Bézier que satisface esta condición es tres.

3) Modelo cinemático del robot

Se considera un robot móvil Pioneer 3DX, tipo uniciclo, detallado y validado experimentalmente en De la CRUZ et al. (2006). Un esquema del robot se muestra en la Figura 3 y se presenta su modelo cinemático en la Ecuación (1).

La posición del robot está definida por el punto $h = [x,y]$, este punto está ubicado a una distancia a desde el centro del eje trasero del robot B. G es el centro de gravedad, E es el punto de

ubicación del sensor Kinect, C es el punto de ubicación de la rueda castor. ω , es la velocidad angular del robot. u y \bar{u} , corresponden a las velocidades longitudinal y transversal del centro de masa del robot. ψ , corresponde al ángulo de orientación del robot.

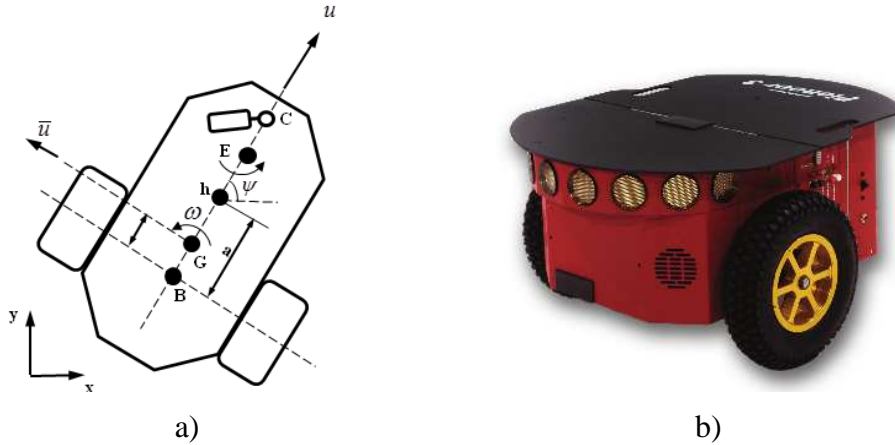


FIGURA 3. a) Puntos y variables de interés sobre el robot. b) Aspecto real del robot.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \\ \dot{\omega} \end{bmatrix} = \underbrace{\begin{bmatrix} u \cos \psi - a \omega \sin \psi \\ u \sin \psi + a \omega \cos \psi \\ \omega \\ \frac{\theta_3}{\theta_1} \omega^2 - \frac{\theta_4}{\theta_1} u \\ -\frac{\theta_5}{\theta_2} u \omega - \frac{\theta_6}{\theta_2} \omega \end{bmatrix}}_{f(x(t))} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{\theta_1} & 0 \\ 0 & \frac{1}{\theta_2} \end{bmatrix}}_{g(v(t))} \begin{bmatrix} u_c \\ \omega_c \end{bmatrix} + \underbrace{\begin{bmatrix} \delta_x \\ \delta_y \\ 0 \\ \bar{\delta}_u \\ \bar{\delta}_\omega \end{bmatrix}}_{\delta} \quad (1)$$

La Ecuación (1) puede escribirse de manera compacta como se muestra en la Ecuación (2).

$$\dot{X}(t) = f(x(t)) + g(v(t)) + \delta \quad (2)$$

$x(t) = [x \ y \ \psi \ u \ \omega]^T$ es el vector de estado del sistema, $v(t) = [u_c \ \omega_c]^T$ es el vector de control y δ , es el vector de perturbaciones o incertidumbre. En otras palabras la Ecuación (2) dice que la variación del estado del robot, es función del estado actual más una función de las acciones de control y de las perturbaciones que actúen sobre él.

Los parámetros δ_x y δ_y dependen del deslizamiento de las ruedas (patinaje) y a la orientación del robot; $\bar{\delta}_u$ y $\bar{\delta}_\omega$ dependen de parámetros mecánicos del robot.

Los parámetros θ del robot móvil son: $\theta_1 = 0.2604 \text{ s}$; $\theta_2 = 0.2509 \text{ s}$; $\theta_3 = -0.000499 \text{ s m rad}^{-2}$; $\theta_4 = 0.99065 \text{ s}$; $\theta_5 = 0.00263 \text{ s m}^{-1}$; $\theta_6 = 1.0768$, identificados en DE LA CRUZ, (2006).

4) Algoritmo de seguimiento de trayectoria mediante la aproximación de Euler.

Utilizando la metodología propuesta por RÓZALES *et al.* (2007), considérese la siguiente ecuación diferencial:

$$y' = f(y, u, t) \text{ con } y(0) = y_0 \quad (3)$$

Donde, y es la salida del sistema a controlar, y' la primera derivada, u la acción de control y t , el tiempo.

Los valores de $y(t)$ en el tiempo discreto $t = kT_o$, serán llamados y_k , donde: T_o representa el período de muestreo, y $k \in \{0, 1, 2, 3, \dots\}$.

Luego, si se desea calcular el valor de y_{k+1} conociendo el valor de y_k , se deberá integrar la Ecuación (4) sobre el intervalo de tiempo $kT_o \leq t \leq (k+1)T_o$.

$$y_{k+1} = y_k + \int_{kT_o}^{(k+1)T_o} f(y, u, t) dt \quad (4)$$

Existen muchos métodos de integración numérica, cada uno de ellos con sus correspondientes algoritmos para realizar el cálculo de y_{k+1} . Por ejemplo, puede hacerse una aproximación con la Ecuación (5).

$$y_{k+1} \cong y_k + T_o f(y_k, u_k, t_k) \quad (5)$$

Donde el valor de y_{k+1} no se conoce, pero puede ser aproximado con la Ecuación (5). En el área de métodos numéricos, esta aproximación se denomina método de Euler. Euler es un método de primer orden que aproxima la solución por la recta tangente (MARTINS et al. 2009).

El uso de los métodos numéricos en el cálculo de la evolución de sistemas, se basa principalmente, en la posibilidad de aproximar el estado del sistema en el instante de tiempo $k+1$, si el estado y la acción de control en el instante de tiempo k son conocidas.

Entonces, el valor y_{k+1} puede ser sustituido por la trayectoria deseada y luego de ello, calcular las acciones de control necesarias para hacer que la salida del sistema, vaya de su valor actual k , a su valor deseado.

a) Modelo del controlador.

Se discretiza el modelo dinámico del robot móvil mediante la aproximación de Euler (Ecuación (3)). De esta forma el modelo dinámico discreto del robot móvil puede ser expresado como se muestra en la Ecuación (6).

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \psi_{k+1} \\ u_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \psi_k \\ u_k \\ \omega_k \end{bmatrix} + T_o \left\{ \begin{bmatrix} u_k \cos \psi_k - a \omega_k \sin \psi_k \\ u_k \sin \psi_k + a \omega_k \cos \psi_k \\ \omega_k \\ \frac{\theta_3}{\theta_1} \omega_k^2 - \frac{\theta_4}{\theta_1} u_k \\ -\frac{\theta_5}{\theta_2} u_k \omega_k - \frac{\theta_6}{\theta_2} \omega_k \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{\theta_1} & 0 \\ 0 & \frac{1}{\theta_2} \end{bmatrix} \begin{bmatrix} u_{ck} \\ \omega_{ck} \end{bmatrix} \right\} + \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \\ \delta_u \\ \delta_\omega \end{bmatrix} \quad (6)$$

Que en términos de Euler puede escribirse como en la Ecuación (7)

$$x_{k+1} = x_k + T_o [f(x_k) + g(v_k)] + \delta \quad (7)$$

Luego se sustituye el vector de estados: $X_{k+1} = [x_{k+1} \ y_{k+1} \ \psi_{k+1} \ u_{k+1} \ \omega_{k+1}]^T$

Por el vector de estados deseados (posición, velocidad, orientación):

$$Xd_{k+1} = [xd_{k+1} \ yd_{k+1} \ \psi d_{k+1} \ ud_{k+1} \ \omega d_{k+1}]^T \quad (8)$$

Si hacemos: $u = [u_{ck} \ \omega_{ck}]^T$,

$$A = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{\theta_1} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\theta_2} \end{bmatrix}^T \quad (9)$$

$$b = \frac{1}{T_0} \begin{bmatrix} x_{dk+1} - x_k \\ y_{dk+1} - y_k \\ \psi_{dk+1} - \psi_k \\ u_{dk+1} - u_k \\ (\omega_{dk+1} - \omega_k) \end{bmatrix} - \begin{bmatrix} u_k \cos \psi_k - a \omega_k \sin \psi_k \\ u_k \sin \psi_k + a \omega_k \cos \psi_k \\ \omega_k \\ \frac{\theta_3}{\theta_1} \omega_k^2 - \frac{\theta_4}{\theta_1} u_k \\ -\frac{\theta_5}{\theta_2} u_k \omega_k - \frac{\theta_6}{\theta_2} \omega_k \end{bmatrix} - \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \\ \bar{\delta}_u \\ \bar{\delta}_\omega \end{bmatrix} \quad (10)$$

Podemos describir la Ecuación (6) como: $Au=b$. La Ecuación (6), es un sistema de 5 ecuaciones con 2 incógnitas y dado que A es una matriz orto normal, ($A^{-1}=A^T$) es posible encontrar una solución optima para u mediante mínimos cuadrados (STRANG 1980, citado por RÓSALES et al., 2009), solución mostrada en Ecuación (11).

$$\begin{bmatrix} u_{ck} \\ \omega_{ck} \end{bmatrix} = \begin{bmatrix} \frac{\theta_1(u_{dk+1}-u_k)-T_0(\theta_3\omega_k^2-\theta_4u_k+\theta_1\delta_u)}{T_0} \\ \frac{\theta_2(\omega_{dk+1}-\omega_k)-T_0(\theta_5u_k\omega_k-\theta_6\omega_k+\theta_2\delta_\omega)}{T_0} \end{bmatrix} \quad (11)$$

Empleando una metodología similar a la utilizada para obtener la Ecuación (11), DE LA CRUZ, (2006) propone que u_{dk+1} y w_{dk+1} pueden ser calculados como mediante la Ecuación (12).

$$\begin{bmatrix} u_{dk+1} \\ \omega_{dk+1} \end{bmatrix} = \frac{1}{T_0} \begin{bmatrix} k_u(\Delta x \cos \psi_k - \Delta y \sin \psi_k) \\ k_\omega \left(\frac{-a \Delta x \sin \psi_k + a \Delta y \cos \psi_k + \Delta \psi}{a^2 + 1} \right) \end{bmatrix} \quad (12)$$

Donde:

$$\begin{aligned} \Delta x &= x_{dk+1} - x_k \\ \Delta y &= y_{dk+1} - y_k \\ \Delta \psi &= \psi_{dk+1} - \psi_k \end{aligned}$$

b) Ajuste de los parámetros del controlador mediante algoritmos genéticos

Como se observa en la Ecuación (12), las velocidades del robot dependen de los parámetros k_u y k_w , los que deben ser ajustados para el mejor seguimiento de la trayectoria. De este modo, si se conoce la trayectoria a seguir, es posible determinar los parámetros k_u y k_w que hagan que el error de seguimiento sea mínimo.

Ya a comienzos de los noventas, diversos autores demostraban (GILL et al. 1995 y KRISTINSSON et al. 1992) las ventajas de usar algoritmos genéticos en la optimización de controladores de trayectorias en robots.

De éste modo, se empleó el *Global Optimization Toolbox* de Matlab, en particular el *Genetic Algorithm solver*. Dicha herramienta se empleó para minimizar la siguiente función objetivo:

$$\text{Mínimo [RMSE}(X_c(t)- X_r(t)) \quad (13)$$

Donde: RMSE, es la raíz cuadrada del error cuadrático medio. $X_c(t)$ es la trayectoria seguida por el robot y $X_r(t)$ es la trayectoria real o trayectoria deseada.

RESULTADOS Y DISCUSION

Se realizaron diversas simulaciones para diferentes trayectorias, con y sin obstáculos, y se midió mediante el RMSE, el error cometido en cada una de ellas.

Primer Escenario. Presentado en la Figura 4, consiste en la búsqueda del camino más corto entre la posición actual del robot (coordenada [1.5, 18.5]) y la maleza más cercana (coordenada [25.5, 13.5]) en un cultivo de 16 hileras, espaciadas a 2 m. El objetivo es alcanzado y la trayectoria de 30 m es generada en 162 ms (Procesador Intel ® Core 2 duo de 1.6 GHz).

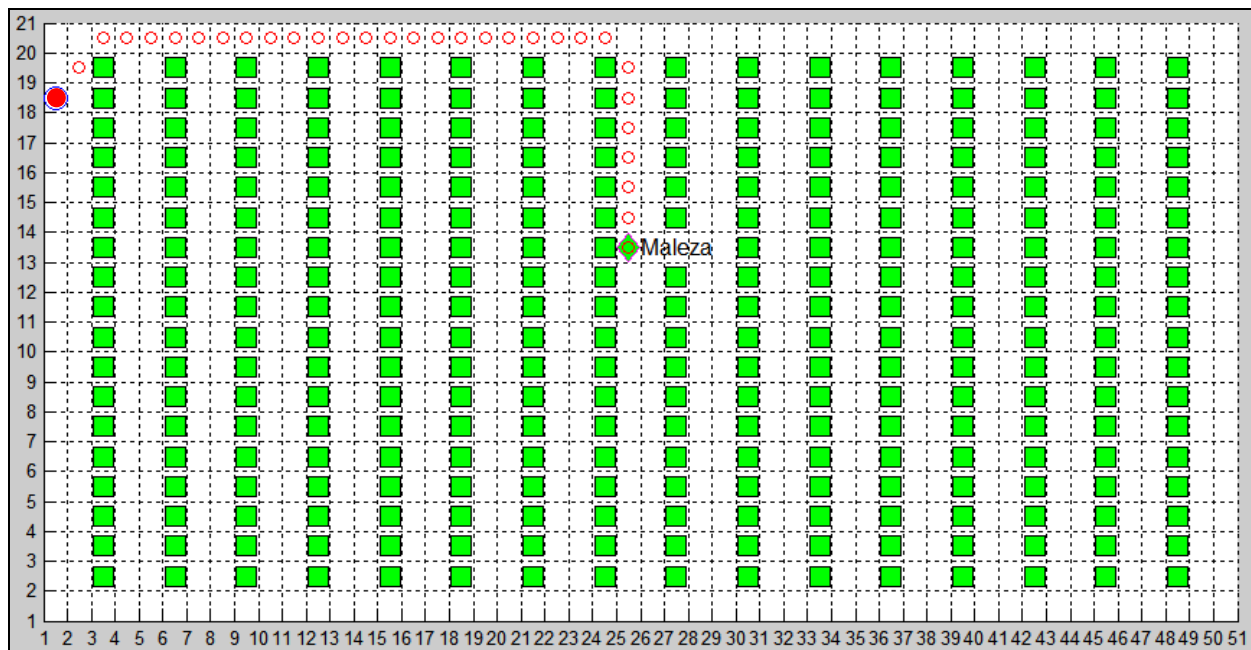


FIGURA 4. Generación de la trayectoria entre las hileras del viñedo simulado (50x20m). Largo trayectoria: 30 m. Entorno de simulación Matlab 2010b.

Segundo Escenario. Mostrado en la Figura 5, presenta un cultivo de 11 hileras, espaciadas a 3 m y contiene 14 obstáculos (cuadrados de color azul). El robot parte de las coordenadas [2.5, 12.5] y debe alcanzar la maleza de las coordenadas [42.5, 4.5]. En este caso la trayectoria de 84.4 m es generada exitosamente en 238 ms.

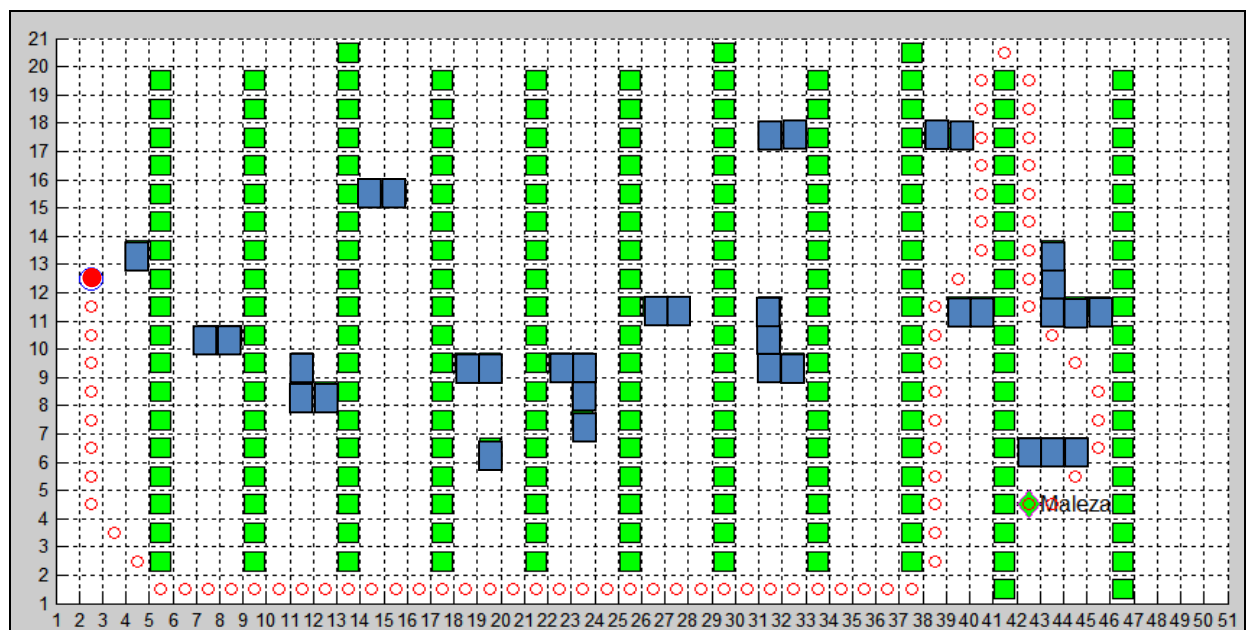


FIGURA 5. Generación de la trayectoria entre las hileras del viñedo simulado (50x20m) con múltiples obstáculos en la ruta. Largo trayectoria: 84,4 m.

Suavizado de trayectorias mediante curvas de Bézier. A modo de ejemplo en la Figura 6 se puede apreciar que la trayectoria generada es suave y permite unir todos los puntos generados en el Primer Escenario. Para ello necesita 340 ms.

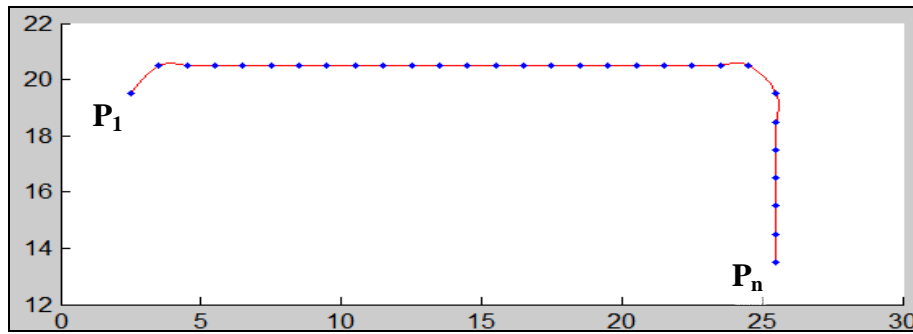


FIGURA 6. Generación de trayectoria suave mediante curvas de B ezier c ubicas.

Controlador de trayectoria. En la Figura 7a, se puede apreciar el desempe o del controlador siguiendo la trayectoria, as  como siguiendo las velocidades deseadas en el primer escenario usando par metros del controlador no optimizados. El error RMSE, es de 0.54 m para una trayectoria de 30 m. Adem s, puede observarse que la velocidad se mantiene oscilando entorno a los 0.4 ms^{-1} , que es mayor a los 0.2 ms^{-1} de velocidad impuesta como referencia.

N tese, que el robot no alcanza su objetivo, quedando a 0.53 m de  sta. En la Figura 7b, se puede observar la desviaci n de la velocidad deseada versus la generada por el controlador.

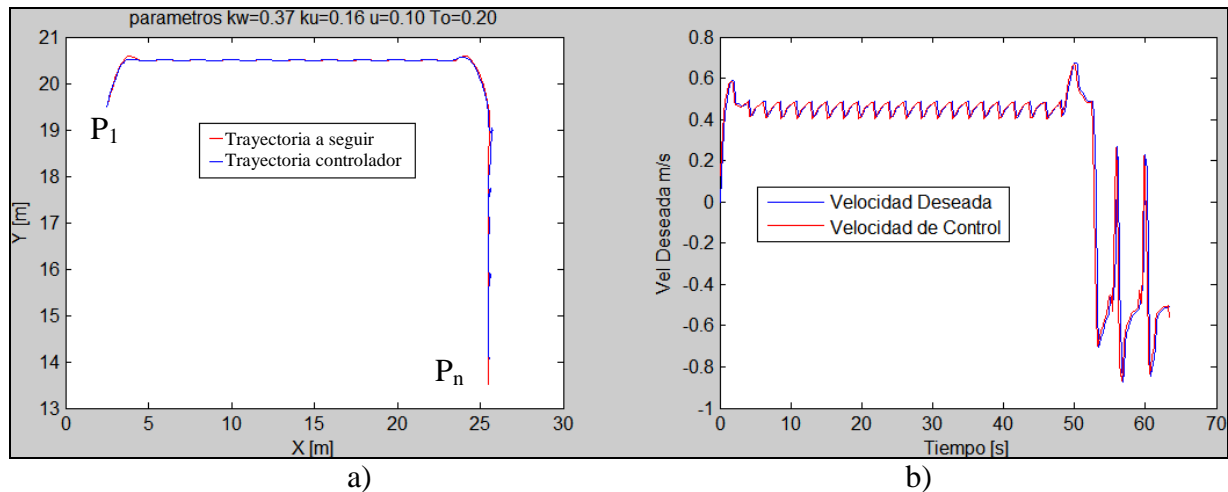


FIGURA 7. Resultados del seguimiento de la trayectoria con par metros K_ω y K_u no  ptimos.

Por otra parte, al usar par metros  ptimos, como se observa en la Figura 8, el robot llega a su destino con un $\text{RMSE}=0.1 \text{ m}$, con velocidades de hasta 1 ms^{-1} .

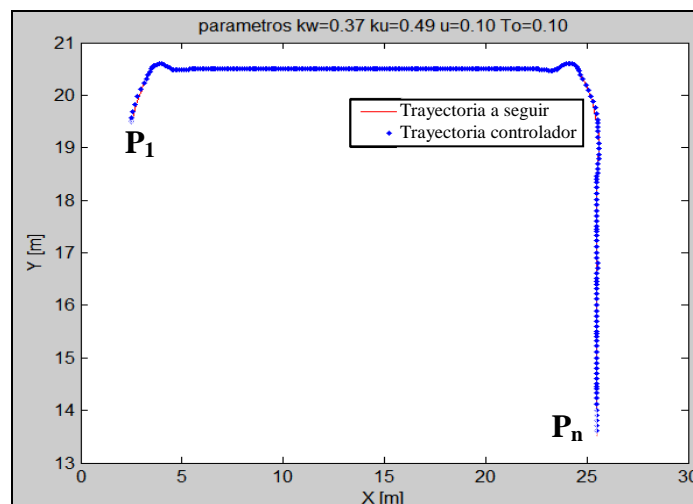


FIGURA 8. Resultados del seguimiento de la trayectoria con par metros K_ω y K_u  ptimos.

Tercer Escenario. Para el caso de detectar un objeto inesperado en el camino, se añadieron en forma aleatoria imágenes 3D de objetos previamente grabados en terreno, como se muestra en la Figura 9.

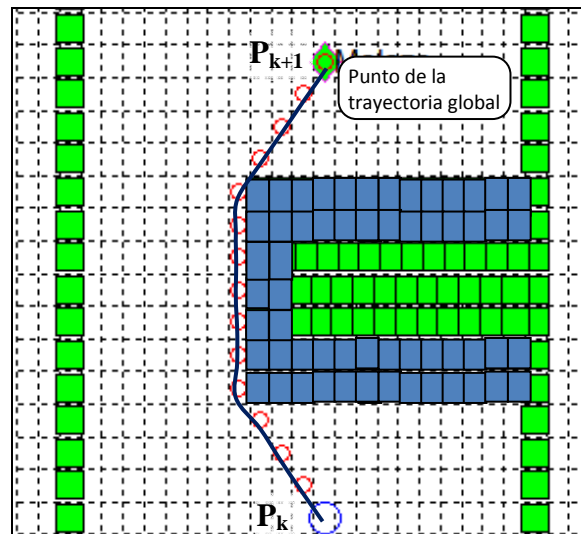


FIGURA 9. Trayectoria para evasión de un obstáculo local y suavizado mediante curvas cúbicas de Bézier. ■ Offset del obstáculo.

De los resultados obtenidos en el todos los Escenarios, podemos decir que, la generación de trayectorias resulta robusta y eficiente en términos de tiempo de computo. Además, para la evasión local de obstáculos resulta una alternativa fiable ya que de haber espacio para que el robot pase, el algoritmo A^* , encontrara la ruta de menor costo. Pero sin embargo, esta ruta tiende a ser muy rectilínea, es decir con aristas pronunciadas, por lo que se hace necesario reducir la velocidad del robot a la mitad (0.1 m s^{-1}) para que el controlador pueda seguir la trayectoria de forma ajustada y evitar así las colisiones en espacios reducidos.

Por otra parte la evasión de obstáculos resulta exitosa, siempre que la altura del objeto sea menor que la altura a la que se encuentra el sensor Kinect. Esto, debido a que si no ve su punto de destino (punto intermedio de la trayectoria global P_{k+1}), el robot necesita generar una nueva trayectoria al siguiente punto intermedio P_{k+2} , lo que no siempre resulta ser la trayectoria de menor costo para evadir un obstáculo.

En cuanto al controlador, este resulta adecuado, ya que considera el modelo cinemático del robot, pero dicha cinemática cambia si es que se le añade peso adicional, como por ejemplo un pequeño estanque con herbicida. Por lo que los parámetros θ deben ser recalculados.

Adicionalmente, el desempeño del controlador está fuertemente ligado a los parámetros K_ω y K_u . Para nuestro análisis los valores seleccionados son los que ofrecen el menor RMSE posible. En el Primer Escenario el RMSE es de apenas 0.1 m, lo que resulta altamente deseable en entornos agrícolas, ya que no posee oscilaciones que le pudiesen provocar una colisión.

Finalmente, el mayor aporte de este trabajo, creemos estriba en la integración de diversas técnicas de planificación, navegación, seguimiento de trayectorias, optimización y evasión de obstáculos mediante visión 3D. Técnicas que permiten asegurar la navegación segura y efectiva, en entornos agrícolas. Combinando técnicas clásicas como A^* , con nuevas herramientas de visión 3D, como la que nos ofrece Kinect.

CONCLUSIONES

El marco de integración de algoritmos de seguimiento y planificación de trayectorias propuesto, muestra bajos errores en el seguimiento de las trayectorias (RMSE=0.1 m en 30 m), debido a la optimización de los parámetros del controlador, para cada trayectoria. Es decir el controlador, genera la mejor solución para cada trayectoria.

Los parámetros del controlador obtenidos mediante algoritmos genéticos garantizan una solución casi óptima, ya que evita caer en mínimos locales a diferencia de otros métodos de optimización.

El ampliamente probado algoritmo de planificación A*, ha probado su eficacia en la generación de trayectorias tanto locales como globales y su eficiencia en términos de tiempo de computo (<300 ms, para la trayectoria más larga). Lo que lo hace adecuado para las necesidades de los robots en entornos agrícolas.

La evasión local de obstáculos detectados por el sensor Kinect, resulta adecuado para la detección de objetos en el camino del robot, aunque estos sean de pequeño tamaño. Pero se debe tener en cuenta en su implementación en terreno, el efecto de la luz sobre el sensor de profundidad y los efectos de la vibración en la determinación de las alturas de los objetos detectados.

REFERENCIAS

- DE LA CRUZ C., "Control de Formación de Robots Móviles", Tesis Doctoral, Instituto de Automática, Universidad Nacional de San Juan, Argentina, San Juan, (2006).
- FLETCHER, L.; TELLER, S.; OLSON, E.; MOORE, D.; KUWATA, Y.; LEONARD, J.; MILLER, I.; CAMPBELL, M. The MIT–Cornell Collision and Why It Happened. 2008.
- GILL, M. & ZOMAYA, A. Genetic algorithms for robot control *Evolutionary Computation*, 1995., *IEEE International Conference on*, , 1, 462.
- HART, P. E.; NILSSON, N. J. & RAPHAEL, B. A Formal Basis for the Heuristic Determination of minimum Cost Paths *Systems Science and Cybernetics, IEEE Transactions on, Systems Science and Cybernetics, IEEE Transactions on, Artificial Intelligence Group of the Applied Physics Laboratory, Stanford Research Institute, Menlo Park, Calif., IEEE*, 1968, 4, 100-107.
- KRISHNA, K., AND G. A. DUMONT, "System identification and control using genetic algorithms", *IEEE Trans on System, Man and Cybernetics*, vol. 22, no. 5, pp. 1033-1046, 1992.
- MARTINS, F.N. SARCINELLI-FILHO, M. BASTOS, T.F. CARELLI, R. Dynamic modeling and adaptive dynamic compensation for unicycle-like mobile robots. *Advanced Robotics*, 2009. ICAR 2009. International Conference on. Junio 2009.
- MATTHEW MCNAUGHTON, CHRISTOPHER R. BAKER, T. G. B. S. C. U. & ZIGLAR, J. ROUFF, C. & HINCHEY, M. (Eds.) *Experience from the DARPA Urban Challenge*, Springer-Verlag London Limited, 2012.
- ROSALES A., G SCAGLIA, V. MULT, "Tracking control of a mobile robot based on Taylor Formula". 2007. *AJIEE. Jornadas en Ingeniería Eléctrica Y Electrónica*.
- ROSALES A., "Dynamic nonlinear model based predictive control for mobile robots". En V Seminario Euro Latinoamericano de Sistemas de Ingeniería. Nov. de 2009.
- SCAGLIA G., MUT V., QUINTERO O., DI SCIASCIO F., "Trajectory Control Of Mobile Robots Based On Numerical Methods", XII Congreso Latinoamericano de Control Automático - CLCA, Salvador, Brasil, (2006a).
- SOFMAN, B.; BAGNELL, J. A.; STENTZ, A. & VANDAPEL, N. *Terrain classification from aerial data to support ground vehicle navigation Robotics*, Citeseer, 2006